

5

## 20

2

3

:

**comprising:**

[storing] lock data representing granted and pending resource lock requests, wherein the data representing each granted and pending resource lock request includes: data indicating a resource to which access has been granted or requested, and an access mode associated with the resource lock request;

resource lock requests are parameterized resource lock requests and the data representing each resource lock request in the subset further includes one or more parameter values indicating a data reliability classification associated with the resource lock request; and

denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request, determining when the resource lock request is conditionally conflicting with any granted resource lock

2. The resource lock management system of claim 1, wherein the resource lock request is conditionally conflicting with a granted resource lock request only when (A) both the resource lock request and the granted resource lock request are parameterized resource lock requests and (B) the resource lock request is neither unconditionally conflicting nor unconditionally compatible with the granted resource lock request.
3. The resource lock management system of claim 2, wherein when resource lock request is a parameterized read lock request  $R(A)$ , where  $A$  represents the parameter values for the parameterized read lock request, and the granted resource lock request is a parameterized write lock request  $W(B)$ , where  $B$  represents the parameter values for the parameterized write lock request, the lock manager determines whether the resource lock request is conflicting with the granted resource lock request by determining whether or not  $B$  is a subset of  $A$ .
4. A resource lock management method,

storing, in a data structure system, lock data representing granted and pending resource lock requests, wherein the data representing each granted and pending resource lock request includes: data indicating a resource to which access has been granted or requested, and an access mode associated with the resource lock request;

evaluating, granting and denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request, determining when the resource lock request is conditionally conflicting with any granted resource lock request, and evaluating the resource lock request with respect to each conditionally conflicting granted resource by performing a predefined comparison of the parameter values for the resource lock request with the parameter values for each conditionally conflicting granted resource lock request,

a first data structure which stores information of a pending or granted lock request, the first data structure including:

a field which stores an identification of a transaction associated with the first data structure; and

a second data structure which stores information of a lock, the second data structure including:

a field which references the first data structure.

5. The resource lock management method of claim 4, wherein the resource lock request is

lock manager means for evaluating, granting and denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request, determining when the resource lock request is conditionally conflicting with any granted resource lock request, and evaluating the resource lock request with respect to each conditionally conflicting granted resource lock request by performing a predefined comparison of the parameter values for the resource lock

a field which references the second data structure.

a field which stores an identification of a most restrictive access mode of the lockable resource and which corresponds to said access mode associated with the resource lock request.

16. A computer program product according to claim 7, wherein the lock data structure system further includes a third data structure, the third data structure including:

a field which references the second data structure.

17. A computer program product according to claim 16, wherein:

the second data structure further includes:

a field which stores aggregated read parameters of first data structures referenced by the second data structure; and

a field which stores aggregated write parameters of first data structures referenced by the second data structure.

wherein the aggregated read and write parameters correspond to said one or more parameter values indicating a data reliability classification associated with the resource lock request.

18. A computer program product according to claim 17, wherein the second data structure further includes:

a field which stores an identification of a most restrictive access mode of the lockable resource and which corresponds to said access mode associated with the resource lock request.

19. A resource lock management system, comprising:

a lock data structure system which stores lock data representing granted and pending resource lock requests, wherein the data representing each granted and pending resource lock request includes: data indicating a resource to which access has been granted or requested, and an access mode associated with the resource lock request;

wherein a subset of the granted and pending resource lock requests are parameterized resource lock requests and the data representing each resource lock request in the subset further includes one or more parameter values indicating a data reliability classification associated with the resource lock request; and

a lock manager for evaluating, granting and denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request.

23. The resource lock management system of claim 19, wherein the resource lock request is conditionally conflicting with a granted resource lock request only when (A) both the resource lock request and the granted resource lock request are parameterized resource lock requests and (B) the resource lock request is neither unconditionally conflicting nor



unconditionally compatible with the granted resource lock request.

24. The resource lock management system of claim 23, wherein when resource lock request is a parameterized read lock request  $R(A)$ , where  $A$  represents the parameter values for the parameterized read lock request, and the granted resource lock request is a parameterized write lock request  $W(B)$ , where  $B$  represents the parameter values for the parameterized write lock request, the lock manager determines whether the resource lock request is conflicting with the granted resource lock request by determining whether or not  $B$  is a subset of  $A$ .

25. A resource lock management method, comprising the steps of:

storing, in a data structure system, lock data representing granted and pending resource lock requests, wherein the data representing each granted and pending resource lock request includes: data indicating a resource to which access has been granted or requested, and an access mode associated with the resource lock request;

wherein a subset of the granted and pending resource lock requests are parameterized resource lock requests and the data representing each resource lock request in the subset further includes one or more parameter values indicating a data reliability classification associated with the resource lock request; and

evaluating, granting and denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request, determining when the resource lock request is conditionally conflicting with any granted resource lock request, and evaluating the resource lock request with respect to each conditionally conflicting granted resource by performing a predefined comparison of the parameter values for the resource lock request with the parameter values for each conditionally conflicting granted resource lock request.

wherein the lock data structure system includes: a first data structure which stores information of a lock, as well as the pending and granted requests thereof, the first data structure including:

a field which stores an identification of a lockable resource which corresponds to said data indicating a resource to which access has been granted

a field which stores an access mode of a resource which corresponds to said access mode associated with the resource lock request;

a second data structure including:

26. A method according to claim 25, wherein the field which stores an access mode stores a most restrictive access mode of the granted lock requests.

a field which stores read parameters; and

**28. A method according to 27, wherein:**

a write parameters are aggregated write parameters of granted write requests.

30. A method according to claim 29, wherein when resource lock request is a parameterized read lock request  $R(A)$ , where  $A$  represents the parameter values for the parameterized read lock request, and the granted resource lock request is a parameterized write lock request  $W(B)$ , where  $B$  represents the parameter values for the parameterized write lock request, the resource lock request is evaluated with respect to the granted resource lock request by determining whether or not  $B$  is a subset of  $A$ .

31. A computer program product for use in conjunction with a computer system, the computer program product comprising a computer readable storage medium and a computer program mechanism

embedded therein, the computer program mechanism comprising:

instructions for storing, in a data structure system, lock data representing granted and pending resource lock requests, wherein the data representing each granted and pending resource lock request includes: data indicating a resource to which access has been granted or requested, and an access mode associated with the resource lock request;

wherein a subset of the granted and pending resource lock requests are parameterized resource lock requests and the data representing each resource lock request in the subset further includes one or more parameter values indicating a classification associated with the resource lock request; and

lock manager means for evaluating, granting and denying resource lock requests, including determining when a resource lock request is unconditionally conflicting with any granted resource lock request, determining when the resource lock request is conditionally conflicting with any granted resource lock request, and evaluating the resource lock request with respect to each conditionally conflicting granted resource lock request by performing a predefined comparison of the parameter values for the resource lock request with the parameter values for each conditionally conflicting granted resource lock request,

wherein the lock data structure system includes:

a first data structure which stores information of a lock, as well as the pending and granted requests thereof, the first data structure including:

a field which stores an identification of a lockable resource which corresponds to said data indicating a resource to which access has been granted or requested; and;

a field which stores an access mode of a resource which corresponds to said access mode associated with the resource lock request;

a field which stores parameters of a data reliability classification associated with a resource lock request which corresponds to said one or more parameter values indicating a data reliability classification associated with the resource lock request;

a second data structure including:

a field which references the first data structure.

a field which stores parameters of a data reliability classification associated with a pending or granted resource lock request;

a memory which stores a second data structure which stores information of a lock, the second data structure including:

a field which stores an identification of a lockable resource; and

a field which references the first data structure;

a memory which stores a third data structure, the third data structure including:

a field which references the second data structure.

38. A resource management system according to claim 37, wherein:

the second data structure further includes:

a field which stores aggregated read parameters of first data structures referenced by the second data structure; and

a field which stores aggregated write parameters of first data structures referenced by the second data structure.

39. A resource management system according to claim 38, wherein the second data structure further includes:

a field which stores an identification of a most restrictive access mode of the lockable resource.

40. A resource management system according to claim 37, wherein the second data structure further includes:

a field which stores an identification of a most restrictive access mode of the lockable resource.

41. A resource lock management system, comprising:

a memory which stores a first data structure which stores information of a lock, as well as the pending and granted requests thereof, the first data structure including:

a field which stores an identification of a lockable resource;

a field which stores an access mode of a resource;

a field which stores parameters of a data reliability classification associated with a resource lock request;

a memory which stores a second data structure, the second data structure including:

a field which references the first data

structure.

42. A system according to claim 41, wherein the field which stores an access mode stores a most restrictive access mode of the granted lock requests.

43. A system according to claim 42, wherein the field which stores parameters comprises:

a field which stores read parameters; and

a field which stores write parameters.

44. A system according to 43, wherein:

the read parameters are aggregated read parameters of granted read requests; and

the write parameters are aggregated write parameters of granted write requests.

09726560-120120